# draft-ietf-sidrops-cms-signing-time

Job Snijders
job@fastly.com

# The goal: optimizing failover from RRDP to RSYNC

**RRDP outages happen from time to time**

- Malformed XML
- Duplicate publish elements for same filename
- TLS errors (for example, certificate expired)
- Unexpected mutation of delta history

**Following draft-ietf-sidrops-prefer-rrdp-02, RPs are expected to switch to RSYNC if there is a problem in RRDP.**

***This failover process can be optimized***

# How RSYNC works (by default)

A given file’s *last modification* timestamp and *filesize* are compared between server and client, in order to decide whether to transfer that given file from server to client.

https://rsync.samba.org/how-rsync-works.html

# RPs ‘naive’ implementations of RRDP

Most RP implementations use POSIX-style filesystem hierarchy as backend.

Most RP implementations don’t consciously set the *last modified* timestamp when writing files out.

Consequently, the *mtime* of all files is ‘now’

# Can’t “recycle” objects from RRDP: mtimes differ

```
$ date
Mon Jul 24 23:31:13 UTC 2023
$ ./src/fort --tal=/etc/rpki/ripe.tal --mode=standalone
$ cd /tmp/fort/repository/chloe.sobornost.net/rpki/uplift/
$ ls -lahtr *.*
-rw-r--r--  1 job  wheel   1.9K Jul 24 23:34 IBfMWA0nPFS6MGTNLNavObgEuIc.mft
-rw-r--r--  1 job  wheel   434B Jul 24 23:34 IBfMWA0nPFS6MGTNLNavObgEuIc.crl
-rw-r--r--  1 job  wheel   1.7K Jul 24 23:34 DyWZUPnQe8WhShqst8ELY4hCABE.roa
-rw-r--r--  1 job  wheel   1.7K Jul 24 23:34 AXG3AfXs41I9eJF3s5OA6mFKqUU.roa
```

*Notice that mtimes on disk differ from what’s on the rsync server:*

```
$ rsync -rt rsync://chloe.sobornost.net/rpki/uplift/
drwxr-xr-x            512 2023/07/24 21:09:28 .
-rw-r--r--          1,704 2023/06/17 00:00:36 AXG3AfXs41I9eJF3s5OA6mFKqUU.roa
-rw-r--r--          1,704 2023/06/17 00:00:36 DyWZUPnQe8WhShqst8ELY4hCABE.roa
-rw-r--r--            434 2023/07/24 21:09:28 IBfMWA0nPFS6MGTNLNavObgEuIc.crl
-rw-r--r--          1,971 2023/07/24 21:09:28 IBfMWA0nPFS6MGTNLNavObgEuIc.mft
```

# Recent rpki-client implementation work

Step 1) When RRDP-originated files are moved from ‘staging’ to ‘valid’, the *last modification* timestamp is changed from UTIME_NOW to:

- thisUpdate (CRLs)
- notBefore (Certificates)
- CMS signing-time (Signed Objects)

Step 2) If a failover to RSYNC is needed, the rsync “—compare-dest” points at the ‘valid’ directory, to only transfer the files that changed.

Step 3) Because the RRDP-originated files have deterministic timestamps, the rsync transfer now is as efficient as possible.

https://github.com/openbsd/src/commit/08ac1330e7955e03b6bba4571e0a9906d7ef5279

# What’s happening in the field?

Many – not all - Certificate Authorities immediately write objects to disk after signing, causing notBefore/CMS signing-time/thisUpdate to “coincidentally” align with the *last modified* timestamps exposed through the rsync service.

If publication point operators start ‘fixing up’ the *last modified* timestamps purposefully, RSYNC efficiency will increase!

# CMS signing-time seems to be a default

man.openbsd.org/CMS_SignerInfo_sign

copy. The *cms* structure will be valid and finalized when this flag is set.

`CMS_PARTIAL`
: If this flag is set in addition to `CMS_REUSE_DIGEST`, the returned *CMS_SignerInfo* structure will not be finalized so additional attributes can be added. In this case an explicit call to **CMS_SignerInfo_sign**() is needed to finalize it.

`CMS_NOCERTS`
: Do not add the signer's certificate to the *certificates* field of *cms*. The signer's certificate must still be supplied in the *signcert* parameter though. This flag can reduce the size of the signature if the signer's certificate can be obtained by other means, for example from a previously signed message.

`CMS_NOATTR`
: Leave the *signedAttrs* field of the returned *CMS_SignedData* structure empty. By default, several CMS *SignedAttributes* are added, including the signing time, the CMS content type, and the supported list of ciphers in an *SMIMECapabilities* attribute.

`CMS_NOSMIMECAP`
: Omit just the *SMIMECapabilities* attribute.

# Updating RFC 6488: mandatory CMS signing-time

Every Signed Objects discovered in the last 12 months carry the CMS signing-time attribute (source: RPKIviews.org)

It seems feasible to mandate CMS signing-time be present, and CMS binary-signing-time be absent, as it wouldn’t put any CA in a state of non-compliance.

# Recommendation

Relying Parties:

- After parsing objects retrieved via RRDP, utimensat() the file's timestamp to align with the relevant internal RPKI timestamp
- Use rsync's –compare-dest feature to recycle and guard your on-disk validated cache

Publication Point Operators:

- Ensure the *last-modified* timestamp of files on the RSYNC server aligns with file's internal RPKI timestamps

# Implementation status

## Relying Party side:

https://www.rpki-client.org/ (version 8.5 and higher)

## Publication side:

https://github.com/job/rpkitouch (a tiny utility to update mtimes)

https://github.com/RIPE-NCC/rsyncit (a RRDP-to-RSYNC converter)

Feedback most welcome

https://github.com/job/draft-sidrops-cms-signing-time

この調子でがんばれ！